ER(Error Recovery)-RTP 라이브러리의 구현

*전승열° **강정구 *최태욱 *정기동
* 부산대학교 전자계산학과
** (주) 미디어 트랜스
{nice10e°, tuchoi, jgkang}@melon.cs.pusan.ac.kr , kdchung@hyowon.cc.pusan.ac.kr

Implementation of ER(Error Recovery)-RTP Library

*Seung-Youl Jun° **Jenog-Gu Kang *Tae-Uk Choi *Ki-Dong Chung
* Dept. of Computer Science, Pusan National University
** Media Trans Technology Inc.

요 약

인터넷상에서의 패킷손실은 멀티미디어의 수신 품질을 크게 저하시킨다. 따라서 멀티미디어 데이터의 실시간성을 유지하면서 손실된 패킷을 복구할 수 있는 에러 제어 기법이 필요하다. 이를 위해 본 논문에서는 어플리케이션 수준에서 패킷단위의 에러제어를 수행하는 RSE 기반 FEC 기법을 사용한다. 또한 FEC 기법의 단점인 부가정보의 양을 네트웍의 상황에 따라 적응적으로 조절하기 위해 본 논문에서는 네트웍의 상태정보(RTP/RTCP)에 따라 부가정보의 양을 조절하는 기법을 제안한다. 그리고 제안한 기법을 실제 어플리케이션에서 쉽게 활용할 수 있게 라이브러리로 구현하였다. 실험에서는 이 라이브러리를 바탕으로 동영상을 전송하는 프로그램을 구현하여 패킷로스에 따른 패킷 복구율과 부가정보의 양을 측정하여, RSE만을 사용하였을 경우와 비교하여 보았다.

1. 서 론

인터넷상에서의 멀티미디어 데이터 전송시 패킷손실은 멀티미디어의 수신 품질을 크게 저하시킨다. 인터넷 전화를 비롯한 영상회의 시스템, 원격 교육 시스템 등과 같은 실시간 데이터 전송과 대화형 인터넷 서비스에서는 이러한 패킷손실이 연속적으로 일어날 경우 수신 품질은 더욱 나빠진다. 따라서 패킷손실에 따른 영향을 최소화하기 위한 기법의 연구가 필요하게 되었다. 이러한 기법에는 크게 전송율 제어(Rate control) 기법과 에러제어(Error control) 기법 등이 제안되었다. 전송율 제어(Rate control) 기법은 네트웍 상태에 따라 패킷의 전송율을 조절함으로써 패킷 손실의 가능성을 줄이는 방법이지만 일반적으로 손실된 패킷을 직접적으로 복구하지는 못한다. 에러 제어 기법은 손실된 패킷을 직접 복구하는 방법으로 대표적으로 재전송(Retransmission) 기법과 FEC(Forward Error Correction) 기법 등이 있다. 재전송 기법은 재전송으로 인한 지연 증가로 실시간 대화식 애플리케이션에는 적합하지 못하다. 그러나 FEC의 경우는 원 정보와 함께 부가정보를 보냄으로써 원 정보가 손실됐을 때 중복정보를 이용해서 손실된 원 정보를 복구한다. 이 기법은 재전송 지연 없이 손실에 대한 탄성을 제공하기 때문에 매우 매력적이다. 그러나 부가정보로 인해 대역폭 오버헤드가 수반되는 단점이 있다. 이러한 단점을 해결하기 위해서 본 논문에서는 RTP/RTCP 프로토콜을 활용하여 네트웍의 상태정보를 획득한후 이에 따라 부가정보의 양을 조절하여, 패킷의 복구율을 최대화 하면서 대역폭의 오버헤드를 최소화 하는 적응적인 FEC 기법을 제안한다. 또한 제안한 기법을 쉽게 어플리케이션에 활용하게 하기 위해 FEC의 대표적인 기법인 RSE와 네트웍의 상태정보를 모니터링해주는 RTP/RTCP를 라이브러리화하였다. 본 논문의 구성은 2장에서는 관련연구인 FEC와 RTP/RTCP 프로토콜에 대해서 설명을 하고, 3장에서는 본 논문에서 제안하는 RSE 기법에 대해 설명하고, 4장에서는 RTP/RTCP를 사용한 RSE의 구현에 대해 설명하고, 5장에서는 실험 및 분석을 통해 제안된 기법의 성능을 평가한 후, 6장에서는 결론 및 향후 연구과제에 대해 기술하였다.

"본연구는 (주)미디어트랜스의 지원으로 수행되었음."

2. 관련 연구

2.1 FEC (Forward Error Correction)

FEC 기법은 패킷이 손실되었을 경우를 대비해 송신측에서 주 정보(main information)와 함께 부가정보(redundant information)을 함께 보내는 방법이다. 수신측에서는 주 정보가 손실되었을 경우 부가정보를 이용하여 주 정보를 복구한다. 즉 FEC에서는 재전송으로 인한 지연이 없이 수신된 정보만으로 효율적으로 패킷 손실을 줄일 수 있다. FEC 기법에서는 부가정보의 양을 처음부터 고정할수도 있지만 네트웍의 상황에 따라 동적으로 변화시키는 것이 가능하다.

2.2 RTP/RTCP

RTP(Real-time transport protocol)는 상호대화식(interactive)의 오디오 또는 비디오와 같은 실시간 데이터의 종점간 전송 서비스를 제공한다. RTP에서는 QoS를 보장하기 위한 메카니즘을 제공하지 않고 RTCP(RTP control protocol)에서 데이터 전송을 모니터하고 세션정보를 전달하여 제어를 수행한다[1].

3. RSE(Reed-Solomon Erasure)기반 적응적 에러 제어

3.1 RSE 코드의 개요

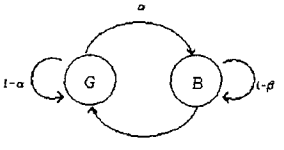
RSE 코드는 잘 알려진 Reed-Solomon 코드에 기반하며 McAuley에 의해 제안되었다[2]. RS 코드가 에러의 위치와 에러 자체를 복구하는 반면, RSE는 에러의 위치를 알고 있는 상태에서 에러 자체만을 정정한다. RSE 인코딩은 RS 코딩에서 사용되는 동일한 polynomial division 방식을 사용해서 수행될 수 있다. 그러나, RSE는 에러 위치를 알고 있는 erasure들만 처리하면 되기 때문에 디코딩 알고리즘이 더 간단하다. 이러한 RSE 코드는 패킷 단위의 에러복구를 위해 애플리케이션 수준에서 많이 이용될 수 있다. 즉, 애플리케이션이 패킷 시퀀스 넘버를 보고 패킷의 손실 여부는 물론 손실된 패킷의 위치를 판단할 수 있기 때문이다.

3.2 RSE 코드의 분석

RSE 인코더는 p 비트 크기의 k개의 데이터 패킷을 이용하여 n-k개의 패리티 패킷을 생성한다. RSE 디코더는 n개의 패킷 중에서 k개의 패킷만을 수신한다면 원래의 데이터 패킷을 복구할 수 있다. 이 때, k개의 데이터 패킷들을 TG(Transmission Group)이라 하고 패리티 패킷들(부가정보)을 포함한 n개의 패킷들을 FEC 블록이라 한다.

RSE(n,k) 코드의 에러복구 능력은 기본적으로 n과 k에 따라 제한된다. n과 k가 고정될 때는 채널의 상태에 영향을 받는다. 즉, 채널의 에러 발생 확률이 클 때 원래의 데이터 패킷을 복구하지 못할 수 있다. 본 논문에서는 채널의 상태에 따라 RSE(n,k)의 성능을 알아보기 위해서 하나의 FEC 블록이 성공적으로 수신될 확률을 계산한다.

각각의 패킷들이 손실될 사건이 독립이 아닐 때 [그림1]의 Gilbert 모델로써 네트웍 상태를 표현할 수 있다[3]. Gilbert 모델의 Good 상태에서는 낮은 확률 P_G로 에러가 발생하고, Bad 상태에서는 높은 확률 P_B로 에러가 발생한다. P_G=0.0이고 P_B=1.0 일 때 Simplified Gilbert Model이라 한다.



[그림 1] Gilbert Model

이 모델에 기반해서 n개중에 최소한 k개의 패킷을 받을 확률 P(n,k)를 계산해 보자. D(n,k)를 n개중에 k개를 받을 확률이고, D(n,k,G)는 n개중에 k개의 패킷을 받은후 G상태에 있을 확률, D(n,k,B)는 n개중에 k개의 패킷을 받은후 B상태에 있을 확률을 나타낸다고 하자. 그러면 D(n,k)=D(n,k,B)+D(n,k,G)가 되며

$$D(n,k,G) = D(n-1,k-1,B)\beta+D(n-1,k-1,G)(1-\alpha) \quad (3-1)$$

$$D(n,k,B) = D(n-1,k,B)(1-\beta)+D(n-1,k,G)\alpha \quad (3-2)$$

로 나타낼수 있다. 이는 boundary condition인 식(3-3)~식(3-5)들을 이용하여 recusively 계산할 수 있다. 여기서 α는 손실 기간의 평균 길이 T_L의 역수이고, β는 비손실 기간의 평균 길이 T_R의 역수이다. T_R과 T_L은 수신측에서 도착하는 패킷들간의 시간간격을 측정함으로써 계산된다.

$$D(n,0,G) = 0 \quad (3-3)$$

$$D(n,n,G) = \pi_G(1-\alpha)^n+\pi_B\beta(1-\alpha)^{n-1} \quad (3-4)$$

$$D(n,n,B) = 0 \quad (3-5)$$

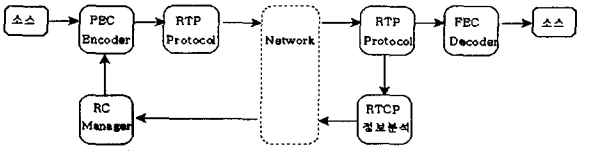
$$D(n,0,B) = \pi_G\alpha(1-\beta)^{n-1}+\pi_B(1-\beta)^n \quad (3-6)$$

π_G는 시스템이 초기에 상태 G에 있을 확률을 가리키며 $\pi_G=\frac{\beta}{\alpha+\beta}$이다. π_B는 시스템이 초기에 상태 B에 있을 확률을 나타내며 $\pi_B=\frac{\alpha}{\alpha+\beta}$이다. 결과적으로 Gilbert 모델에서 n개중에 최소 k개를 받을 확률 P(n,k)의 값은 다음과 같이 나타낼 수 있다.

$$P(n,k) = \sum_{i=k}^{n} D(n,i) \quad (3-7)$$

3.3 RSE 기반 적응적 에러 제어

FEC 기법의 단점은 부가정보의 양만큼 네트웍 대역폭을 요구한다는 점이다. 따라서 네트웍의 상태에 따라 부가정보의 양을 조절해야 한다.



[그림 2] 부가정보 제어를 위한 시스템 구조

RSE(n,k) 코드를 이용하여 부가정보를 조절할 때는 네트웍 상태정보를 이용하여 두 개의 파라메터 n과 k를 주기적으로 결정해야 한다. [그림 2]는 부가정보 제어를 위한 시스템 구조를 보여준다. RTP protocol을 통하여 수신된 패킷들의 정보는 RTCP 정보분석 모듈을 통하여 송신측으로 피드백 된다. FEC 블록의 수신확률 P(n,k)를 구하기 위해서 피드백 정보는 손실 기간의 평균길이 T_L과 비손실 기간의 평균길이 T_R을 포함해야 한다. 이 값들은 매 RTCP주기마다 결정되며 급격한 증감을 방지하기 위해 식(3-8)과 식(3-9)를 이용하여 평활화한다.

$$(T_R)_i = \alpha*T'_R+(1-\alpha)*(T_R)_{i-1} \quad (3-8)$$

$$(T_L)_i = \alpha*T'_L+(1-\alpha)*(T_L)_{i-1} \quad (3-9)$$

RS 코더의 심볼 크기 m은 8또는 32일 때 가장 효율이 좋고, 심볼 크기 P가 m보다 큰 경우 P는 m의 배수이어야 한다[4]. 따라서 패킷 크기 S_{packet}를 결정할 때 8의 배수나 32의 배수를 선택한다. 선택된 S_{packet}를 이용하여 RSE 코더의 TG 크기 k를 식(3-10)에 의해 결정한다.

$$k = \frac{S_{frame}}{S_{packet}} \quad (3-10)$$

k 값이 결정되면 식(3-7)을 이용하여 n을 결정한다. 수신측에서 허용하는 FEC 블록의 손실확률을 τ라 할때 FEC 블록이 $1-\tau$보다 더 큰 확률로 수신측에서 복구될 수 있도록 P(n,k)

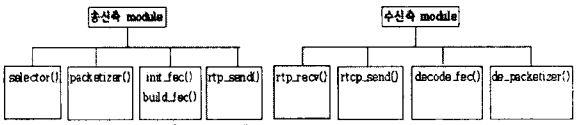
≥ $1-\tau$ 를 만족하는 최소의 n을 선택한다.

$$n = \min \{ t \mid P(t,k) > 1-\tau \},\ k \le t \le \infty \qquad (3\text{-}11)$$

P(n,k)값은 상태천이확률 α, β에 따라 달라지며 이값은 RTCP 주기마다 피드백되는 T_R과 T_L에 의해 주기적으로 변경된다.

4. RTP with RSE의 구현

위 제안된 기법에 따라 구성한 프로그램 모듈도는 아래 [그림 3]과 같다.



[그림 3] 프로그램 모듈도

4.1 module 주요 함수

4.1.1 송신측 module

- selector(float a, floatb) : 상태천이 확률 a,b에 의거해 식(3-1)~식(3-11)을 이용하여 n,k를 정한다
- packetizer(int n, int k) : 정해진 n,k에 따라서 패킷타이제이션 한다.
- init_fec() : fec를 하기위한 초기화
- build_fec(unsigned int *data[], int k, int sz, gf *fec, int index) : 정보패킷(data[])을 encoding 한다

  data 데이타 패킷에 대한 포인터
  k 정보패킷의 개수
  fec 인코딩된 패킷에 대한 포인터
- rtp_send(char *buf, int len) : RTP 패킷으로 만들어 전송한다.

4.1.2 수신측 module

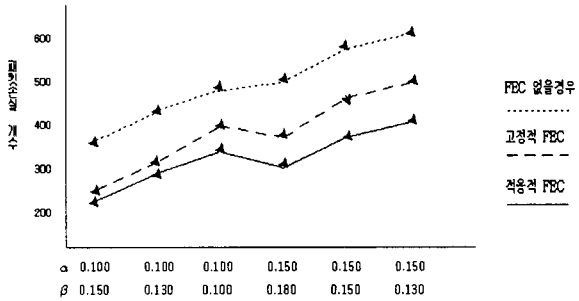
- rtp_recv(char *buf, int len) : RTP 패킷들을 받아서 헤더 정보를 분석한다.
- rtcp_send(u_short sn, u_long ts) : 수신된 rtp 헤더를 분석하여 Sequence Number(sn)과 timestamp(ts)를 바탕으로 패킷 손실기간의 평균길이T_L과 비손실 기간의 평균길이T_R를 구한다. RTCP 확장 필드에T_L과 T_R을 담아 전송한다.
- decode_fec(unsigned int *src[], unsigned int *dst[], int index[], int k, int sz) : 수신된 패킷들(src[])을 decoding하여 원래의 정보패킷(dst[])으로 생성한다.

  src 수신한 패킷에 대한 포인터
  dst 디코딩 된 패킷에 대한 포인터
  index 수신된 패킷의 인덱스 (0. .k-1: 정보패킷; k. .n-1:인코딩된패킷)
  sz 각 패킷의 size
- de_packetizer()

  복구된 패킷들을 원래의 source로 회복한다.

5. 실험 및 분석

실험을 위해 실제 동영상을 전송하는 프로그램을 작성하였다. 에러 모델은 two-state Markov 모델을 사용하였다. Markov 모델은 미래의 상태는 현재의 상태에 의해서만 결정되며, 과거와 독립적인 것으로 간주한다. Markov 모델의 에러 발생율을 조절하여 실험을 하였다. 실험결과는 아래 [그림4]와 같다.



[그림 4] 실험결과

실험결과 FEC를 적응적으로 사용하였을 경우 복구후 패킷 손실개수가 더 작은걸 확인 할수 있다.

6. 결론 및 향후 연구과제

본 논문에서는 패킷손실에 대한 복구 방법으로 네트웍의 상태 정보에 기반한 적응적인 RSE기법을 제안하였고, 어플리케이션 수준에서 활용하기 위해 이를 라이브리화 하였다. 그리고 이 라이브러리를 활용하여 실제 어플리케이션 프로그램을 작성하여 실험을 하였다.

향후 과제로는 이 기법의 모바일 환경에서의 적용가능성과 Rate control 기법과 함께 적용하는 방법에 대해 연구 하고자 한다. 그리고 codec과 어플리케이션의 특징에 맞게 이 기법을 확장하려고 한다.

참고 문헌

[1] H. Schulzrinne, RFC1889(RTP: A Transport Protocol for Real-Time Applications, Audio-Video Transport Working Group 1995.

[2] A.J. McAuley, "Reliable broadband communications using a burst erasure correcting code", Proceedings of ACM SIGCOMM'90, Philadelphia, PA, Sept. 1990

[3] J.R. Yee, E.J. Weldon, "Evaluation of the Performance of Error-Correcting Codes on a Gilbert Channel", IEEE Transactions on Communications, Vol.43 No.8, Aug. 1995.

[4] J. Nonnenmacher, E.W. Biersack, D. Towsley,"Parity-based Loss Recovery for Reliable Multicast Transmission", IEEE/ACM Transactions on Networking, Vol.6 No.4, Aug. 1998.